



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Edificació
de Barcelona

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO DE FIN DE MÁSTER

ESTUDIO DE LAS ETAPAS DE DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA DE SUPERVISIÓN PARA PROCESOS INDUSTRIALES

Proyectista: Pau Martínez Ruiz

Director/es: Miguel Delgado Prieto

Jose Luis Romeral Martín

Convocatoria: Setiembre 2019

Documento I - Memoria

Este estudio trabaja la implementación de una tecnología de modelo de novedad sobre un proceso existente, para ello, se evalúa la información disponible en máquina y las herramientas necesarias para almacenar y trabajar con los datos recolectados.

This study aims to implement novelty model technology on an already existing process, in order to achieve that, the machine is evaluated as to identify what information is available and which tools are required in order to storage and work with the collected data.

AGRADECIMIENTOS

Este trabajo no habría sido posible sin el valioso soporte de Ángel Fernández y Lucía Ruiz, quiero aprovechar este espacio para agradecer enormemente vuestra implicación y ayuda, sois dos grandes profesionales y espero y os deseo lo mejor tanto en lo personal como lo profesional.

También sería inexcusable no agradecer a Miguel Delgado, su apoyo durante todo el desarrollo de este trabajo de fin de master, siempre has estado para ofrecer una solución cuando todo parecía dirigir a un estrepitoso fin.

Por último, pero no menos importante, agradecer a mi padres y hermano su soporte incondicional y continuo desde me alcanza la memoria.

SUMARIO

1	GLOSARIO	7
2	INTRODUCCIÓN	8
2.1	Objetivo.....	8
2.2	Alcance	8
2.3	Requerimientos.....	10
2.4	Justificación	11
3	NÚCLEO DE LA MEMORIA.....	12
3.1	Estado del arte.....	12
3.2	Visión de proyecto y arquitectura	14
3.3	Adquisición y almacenado.....	17
3.3.1	Node-Red.....	17
3.3.2	Servidor local InfluxDB	18
3.3.3	Dispositivo IoT.....	19
3.4	Acondicionamiento y procesado de datos	21
3.4.1	Cálculo de tiempos.....	21
3.4.2	Modelo de novedad.....	22
3.5	Visualización	26
3.5.1	Preparación.....	26
3.5.2	Dashboards.....	27
4	CONCLUSIONES	30
5	BIBLIOGRAFÍA	32

SUMARIO DE FIGURAS

Figura 3.1 Ejemplo de arquitectura de comunicación en máquina	13
Figura 3.2 Jerarquía y flujo de información	14
Figura 3.3 Arquitectura de adquisición de datos	15
Figura 3.4 Flujo de datos entre InfluxDB, Matlab y Grafana.....	16
Figura 3.5 Nodo de lectura Modbus/TCP	17
Figura 3.6 Nodo de escritura InfluxDB	17
Figura 3.7 Flow de adquisición PROFIBUS y CANopen	18
Figura 3.8 Dispositivo IoT basado en Arduino	19
Figura 3.9 Pinzas amperimétricas del dispositivo IoT	19
Figura 3.10 Montaje para medición del motor del conveyor.....	20
Figura 3.11 Modelo de novedad. Representación de la evaluación de medidas 1	23
Figura 3.12 Placa de identificación del motor tractor del conveyor	24
Figura 3.13 Modelo de novedad. Representación de la evaluación de medidas 2	25
Figura 3.14 Definición de DB en Grafana	26
Figura 3.15 Dashboard de mantenimiento y producción	27
Figura 3.16 Registro de diagnósticos del modelo de novedad.....	27
Figura 3.17 Registros de tiempos del proceso.....	28
Figura 3.18 Contador de piezas.....	28
Figura 3.19 Histórico de consumos del motor tractor del conveyor.....	29
Figura 3.20 Dashboard de consumo.....	29

1 GLOSARIO

- API: Acrónimo de Application Programming Interface, es una herramienta asistente en la estructuración de información para su correcto intercambio entre softwares.
- Dashboard: Se trata de una interfaz gráfica donde fácilmente pueden encontrarse datos relevantes para un objetivo o proceso.
- Drive: las referencias a drives en este trabajo, hacen alusión a controladores de motores.
- Fieldbus: En castellano bus de campo, es un protocolo de comunicación entre controladores y equipos industriales a nivel de proceso
- IA: Acrónimo de Inteligencia Artificial, en el ámbito del trabajo cuando hablemos de IA haremos referencia a tecnología capaz de emular la inteligencia cognitiva humana.
- OEE: Acrónimo de Overall Equipment Efficiency, se trata de un indicador matemático que representa en %, la eficiencia productiva de un equipo o máquina.
- Push/Pull: Las referencias a push/pull en el contexto de este trabajo hacen referencia al inicio del flujo de información, siendo push información enviada por “iniciativa propia” y pull información enviada a partir de una petición externa.
- RIO: Acrónimo de Remote Input Output, llamamos RIO a la periferia de señales de campo, englobando señales de entradas como salidas.
- TPM: Acrónimo de Total Productive Maintenance, se trata de una metodología de mejora cuyo objetivo es garantizar la disponibilidad de máquinas y procesos.
- WAN: Acrónimo de Wide Area Network, las menciones a WAN en este trabajo puede interpretarse como menciones a Internet, como red global.

2 INTRODUCCIÓN

2.1 Objetivo

El objetivo de este trabajo es implementar, en un caso de estudio práctico, una solución de supervisión basada en inteligencia artificial orientada al mantenimiento predictivo.

Para ello, partiendo de un proceso productivo funcional existente, se registrarán datos de máquina haciendo uso de la herramientas y dispositivos de adquisición necesarios. Estos datos serán acondicionados para su procesado y posterior utilización en la monitorización del proceso y asistencia en la toma de decisiones.

2.2 Alcance

Con fin de implementar tal solución de supervisión, el desarrollo del trabajo presenta tres etapas diferenciadas: adquisición de datos, acondicionamiento y procesado de datos y, por último, visualización.

Cada una de las etapas, es definida por una lista de tareas independiente que debe ser completada para proceder a etapas posteriores y por ende el resultado final. A continuación, se encuentra el desglose por etapas:

Adquisición de datos

- Definición de la arquitectura y flujo de información planeado
- Integración de las herramientas a nivel de campo
- Integración entre nivel de campo y base de datos

Acondicionamiento y procesado de datos

- Conversión de las medidas recogidas en información procesable
- Entrenamiento de un modelo de novedad
- Comprobación del funcionamiento del modelo

Visualización

- Creación de un dashboard como resultado final
- El dashboard deberá ser capaz de mostrar el histórico de la información recogida
- El dashboard debe ser capaz de mostrar los resultados del procesado de información por el modelo de novedad

Queda totalmente fuera del alcance de este proyecto áreas de trabajo como son:

- Seguridad informática dentro de la infraestructura, englobando así el campo de sistemas relacionado con IT. Conceptos como Firewalls, puertos y encriptación de la información no serán abordados.
- Describir el conjunto de componentes de la máquina, así como como su secuencia y su respectiva programación, tal información puede encontrarse en los trabajos consultado Manufacturing Process automation by means of PLC's under the Industry 4.0 framework¹ y Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de pieza².
- Detallar el funcionamiento del dispositivo IoT de medida del motor tractor de la línea en el caso de estudio.

¹ Vercruysse,J. Manufacturing Process automation by means of PLC's under the Industry 4.0 framework. Trabajo de fin grado, UPC, 2019

² Pumares Cerezo,J. Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de pieza. Trabajo de fin de grado, UPC, 2019

2.3 Requerimientos

Puesto que este caso de estudio práctico se trabaja sobre el trabajo previo de otros estudiantes, así como dentro de un entorno académico, existen

- Deberán usarse herramientas Open-Source o que se encuentren dentro del acuerdo de distribución de software para el alumnado UPC.
- No se modificará en ningún momento la secuencia del proceso, ni la instalación de componentes en máquina.
- El resultado no debe proceder al paso de la toma de decisión automatizada.

2.4 Justificación

A fecha de publicación de este trabajo, nos encontramos en un período de transición entre estándares industriales donde cambiarán los campos del diseño, producción y venta.

Este movimiento viene de la mano de la que se postula como futuro del estado del arte, Industria 4.0, un concepto de industria que aspira a convertir las factorías, y por ende sus procesos y producción, en entidades inteligentes, flexibles y eficientes.

Para lograr un objetivo tan ambicioso, tomarán roles clave, entre otras, tecnologías como Digital Shadow, Digital Twin, Inteligencia Artificial, Internet of Things, realidad aumentada, realidad virtual y robótica colaborativa.

De entre las tecnologías citadas anteriormente, la aplicación de inteligencia artificial al campo del mantenimiento en procesos productivos, supone una mejora en la monitorización del mantenimiento, llegando a poder prever fallos antes de que ocurran, concepto de la mano de metodologías de mejora como TPM.

Una IA que genera información con tal valor añadido, capacita a los responsables del mantenimiento para tomar decisiones informadas con efectos visiblemente positivos sobre la producción, ya que, evitando errores o fallas, incrementa la disponibilidad de máquina.

Además de generar mejoras apreciables y medibles en comparación a otras tecnologías, integrar IA es viable tanto en nuevos sistemas como aquellos ya existentes, por lo general, sin grandes cambios en máquina, ni grandes inversiones, siendo el peor de los casos, aquel en que sea necesario añadir dispositivos de medición.

Teniendo los factores previos en cuenta, existen indicativos de que la IA aplicada al mantenimiento es una tecnología con cabida en el ámbito industrial, y por ello, en anticipación a la estandarización de tales soluciones en el mercado, como estudiante trabajando en el campo de automatización, integrar tales conceptos en un estudio práctico plantea un trabajo enriquecedor.

3 NÚCLEO DE LA MEMORIA

3.1 Estado del arte

De cara a facilitar la interpretación de contenidos y ubicar al lector, en este apartado, se presentan brevemente las herramientas usadas reconocidas en industria, así como que

- *Node-Red*

Node-Red es una herramienta de programación open-source, cuyo fin es facilitar la interconectividad entre equipos físicos y/o virtuales.

La programación es llevada a cabo con Flows, códigos visuales donde se establecen flujos de información entre nodos. Un nodo puede ser bien un punto de entrada o salida de información del código o bien un paso intermedio de acondicionamiento de la misma.

Además de tratarse de software open-source, Node-Red destaca por su fácil integración en dispositivos y por su versatilidad, ya que dispone de una comunidad activa que trabaja, continuamente, en el desarrollo de nuevas librerías para interpretar distintas tecnologías de comunicación, así como librerías con APIs de otros softwares.

- *InfluxDB*

InfluxDB es un gestor de bases de datos indexadas históricamente.

Esta herramienta open-source, se caracteriza por soportar grandes peticiones de escritura y lectura, puntos críticos tanto en la adquisición y posterior tratamiento de datos.

InfluxDB pone a disposición del usuario las herramientas para crear el servidor, local o online, donde alojar los bancos de datos, así como el cliente para interactuar con las mismas.

- *Grafana*

Grafana es un entorno de visualización y creación de dashboards open-source.

De cara a la visualización de información, Grafana pone a disposición del usuario distintas formas de visualizar la información, así como un repositorio con alternativas para personalizar la apariencia.

Grafana también es capaz de integrar la notificación de alertas a tiempo real, así como aplicar lógica sobre la información leída.

▪ *Arquitectura de comunicación en máquina estándar*

En una arquitectura de máquina común, encontramos un PLC y sus dispositivos esclavos bajo su fieldbus. El ejemplo de la figura 2.1 ilustra como tales: controladores para motores, periferia de señales y terminales de válvulas.

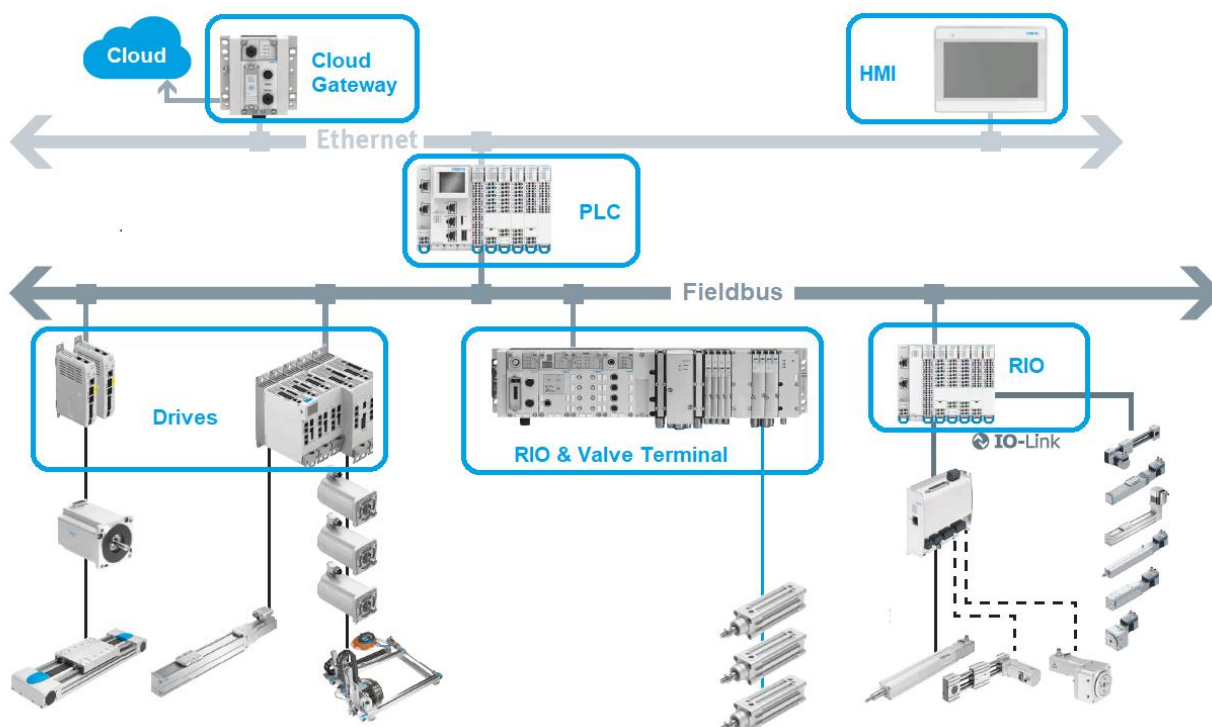


Figura 3.1 Ejemplo de arquitectura de comunicación en máquina³

En un nivel superior al PLC, podríamos encontrar además de los componentes como HMIs o pasarelas, también ordenadores industriales alojando SCADAs o servidores locales de bases de datos entre otros.

En la mayoría de los casos, la red Ethernet corriendo sobre los controladores de un proceso productivo es local, sin acceso directo a la WAN, y en aquellos casos donde es necesario sacar información de la red local, se hace mediante procesos de encriptación de información.

³ Adaptación de: Festo. Electric Automation, pág.3-4. Catálogo Promocional, 2019/06. Obtenido de https://www.festo.com/net/SupportPortal/Files/652002/ElectricAutomation_en.pdf

3.2 Visión de proyecto y arquitectura

Con fin de proceder al desarrollo de la adquisición y pasos posteriores del proyecto, es necesario entender cuál es la situación del caso estudiado.

El caso de estudio, ofrece una línea capaz de trabajar en bucle controlada por dos PLCs y también un analizador de red, para monitorizar sus consumos eléctricos.

Además de esto, para tomar medidas del motor tractor de los conveyors de la línea, se tiene acceso a las lecturas de un dispositivo IoT basado en Arduino que registra los corrientes de fase este.

Con estos recursos, en cuanto a integración, se plantea la siguiente jerarquía, figura 3.2, donde se diferencia entre los componentes de campo, la gestión de la información y la visualización por parte del usuario final.

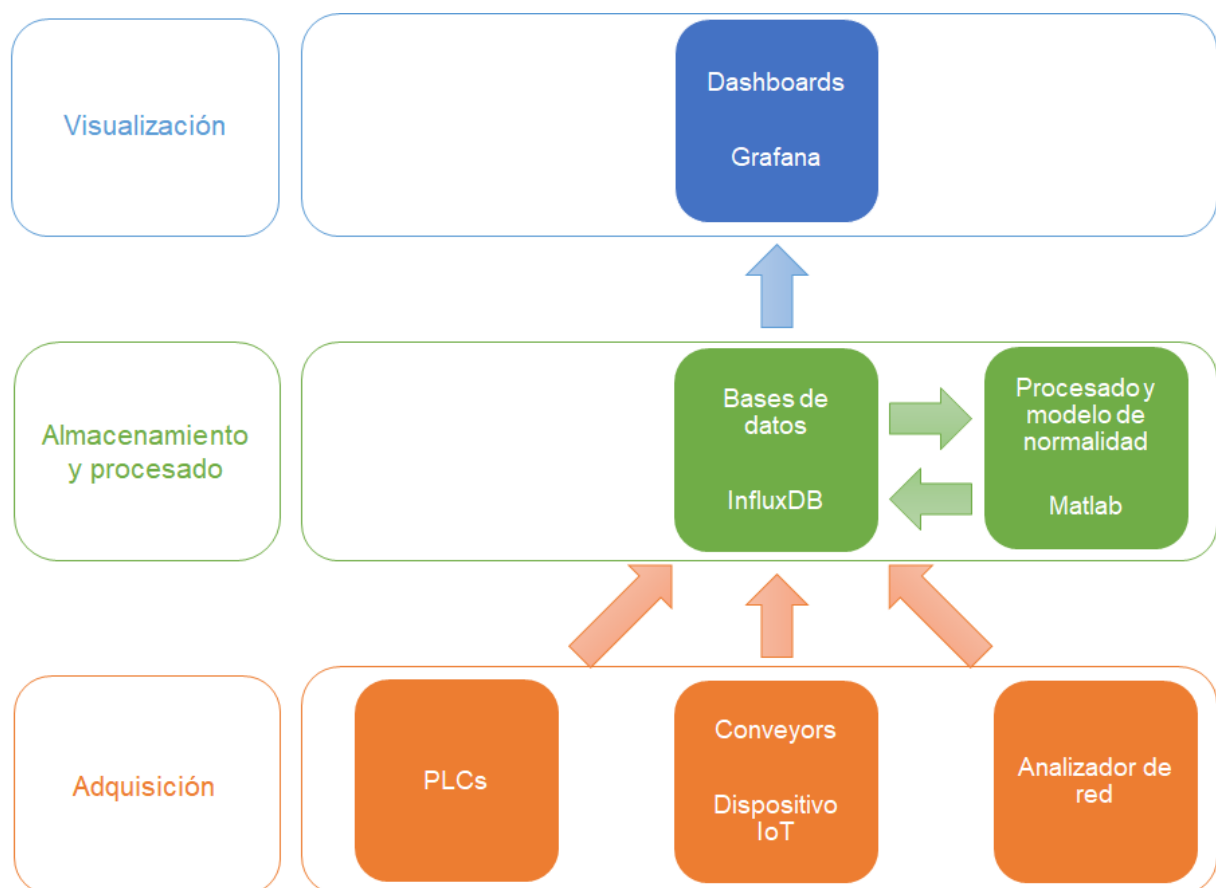


Figura 3.2 Jerarquía y flujo de información

Contrariamente al ejemplo de arquitectura de comunicación común, expuesto previamente, los PLCs del caso de estudio, conectan directamente a la WAN vía Modbus/TCP, de modo que se puede acceder a ellos siempre que se disponga conexión a internet.

Con esto, la arquitectura de adquisición de datos propuesta, es la ilustrada a continuación, figura 2.3.

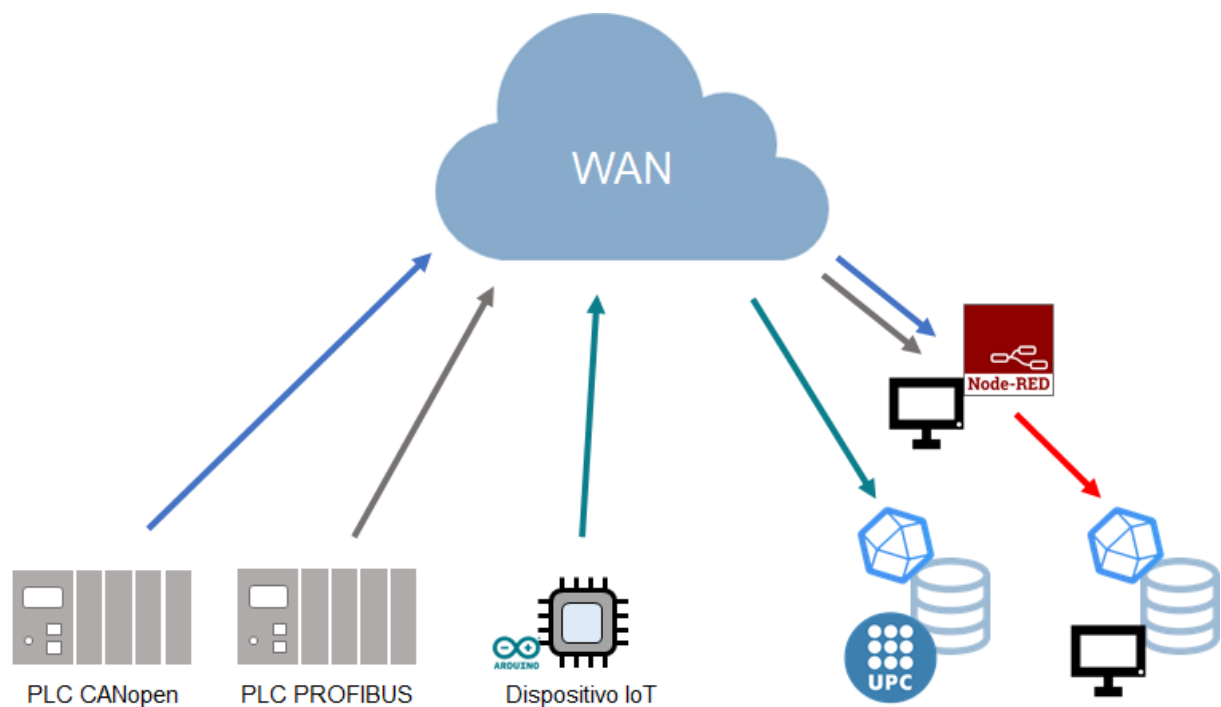


Figura 3.3 Arquitectura de adquisición de datos⁴

En la arquitectura propuesta, una instancia local de Node-Red realizará un pull de información sobre los PLCs vía Modbus/TCP, esta se acondicionará para el posterior push a la base de datos InfluxDB localizada en el servidor local del mismo equipo.

⁴ Icono base de datos procedente de FreelconsLibrary, obtenido de <https://icon-library.net/icon/data-source-icon-22.html>

Icono monitor, obtenido de Noun Project

Logo de Node-Red, obtenido de Node-Red.org

El dispositivo IoT, lleva a cabo la lectura de datos, acondicionamiento de los mismos para guardarlos en InfluxDB y por último el push de contra la base de datos InfluxDB localizada en el servidor UPC GAIA.

Por último, el flujo de datos propuesto una vez la información ha sido registrada en la base de datos es el de la figura 2.4.

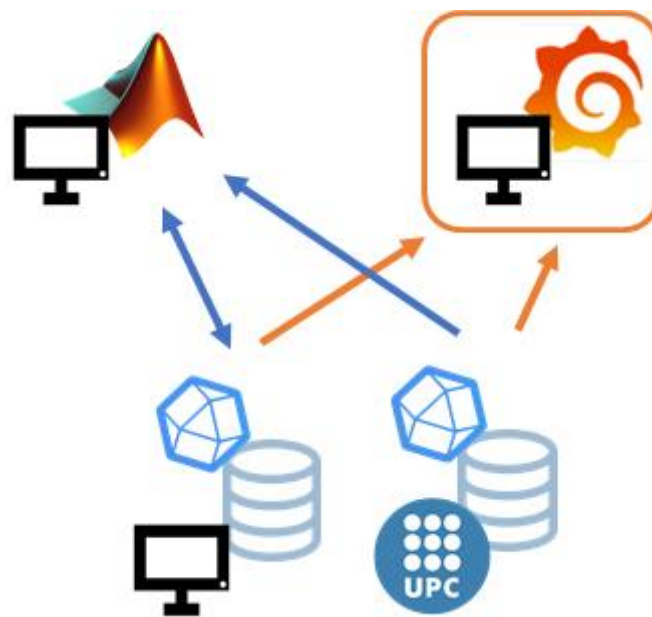


Figura 3.4 Flujo de datos entre InfluxDB, Matlab y Grafana⁵

Se pretende que la información de ambos servidores sea accesible tanto por el entorno de visualización Grafana, como el entorno de cálculo Matlab, donde se procesará el modelo de novedad y los cálculos de tiempos del proceso.

⁵ Icono Matlab, obtenido de Wikipedia,
https://es.wikipedia.org/wiki/MATLAB#/media/Archivo:Matlab_Logo.png

3.3 Adquisición y almacenado

3.3.1 Node-Red

En este subapartado, se presentan las librerías usadas y el Flow final usado para adquirir la información de los estados de proceso de los PLCs.

El código comentado de los nodos, así como el código para importarlo puede encontrarse en los anexos.

3.3.1.1 Librería Modbus/TCP

La librería *node-red-contrib-modbus*⁶ permite acceder a los registros de memoria de los PLCs diferenciando entre bits y words.

Para ello, esta librería dispone de dos nodos, lectura y escritura. En el caso de estudio, se trabaja con el nodo de lectura. En los nodos de lectura se almacenará la información relacionada con dirección del PLC, área de memoria y registros que leer y la frecuencia de muestreo.

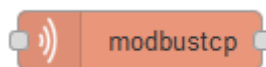


Figura 3.5 Nodo de lectura Modbus/TCP

3.3.1.2 Librería InfluxDB

La librería *node-red-contrib-influxdb*⁷ es capaz de mandar peticiones de escritura al cliente InfluxDB.

El nodo usado de la librería es *influx batch*, el nodo almacena la información de acceso al cliente InfluxDB, así como la base de datos a usar y las credenciales de acceso.

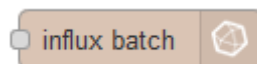


Figura 3.6 Nodo de escritura InfluxDB

⁶ Node-Red.org . node-red-contrib-modbus tcp. 2018.
Obtenido de <https://flows.nodered.org/node/node-red-contrib-modbus tcp>

⁷ Node-Red.org . node-red-contrib- influxdb. 2019.
Obtenido de <https://flows.nodered.org/node/node-red-contrib-influxdb>

3.3.1.3 Flows

El Flow final usado en Node-Red, es presente en la figura 2.7.

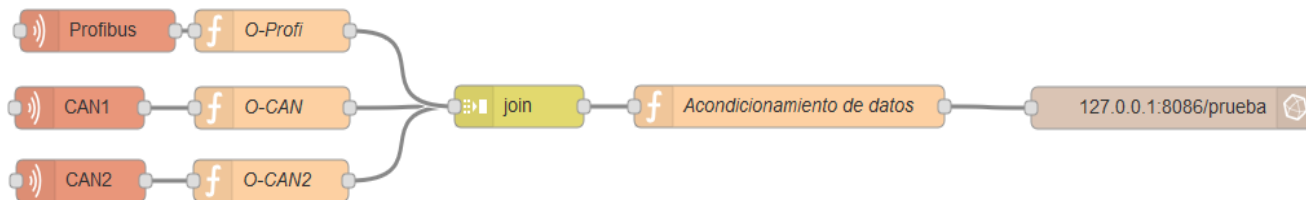


Figura 3.7 Flow de adquisición PROFIBUS y CANopen

Además de los nodos Modbus/TCP de lectura e InfluxDB de escritura, el Flow de adquisición incluye los bloques de función java empezados por “O-“ para sincronizar y organizar la información que envían los nodos Modbus/TCP de lectura.

Una vez la información leída está en orden, un segundo bloque de función java, “Acondicionamiento de datos”, indexa los valores con un título reconocible y les asigna un valor temporal histórico.

También es usado el Flow de adquisición del analizador de red usado en *Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de piezas*⁸, puede encontrarse consultándolo.

3.3.2 Servidor local InfluxDB

La carga de trabajo asociada a la puesta en servicio del servidor local InfluxDB para la adquisición de datos contempla la instalación de servidor, cliente y creación de la base de datos donde almacenar las distintas tablas de datos leídas desde Matlab y Grafana.

⁸ Pumares Cerezo, J. Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de pieza. Trabajo de fin de grado, UPC, 2019

3.3.3 Dispositivo IoT

El dispositivo de adquisición de señales sobre el motor tractor del conveyor, figura 3.8 y 3.9, está basado en Arduino y hace uso de pinzas amperimétricas para leer intensidades de las fases del motor.

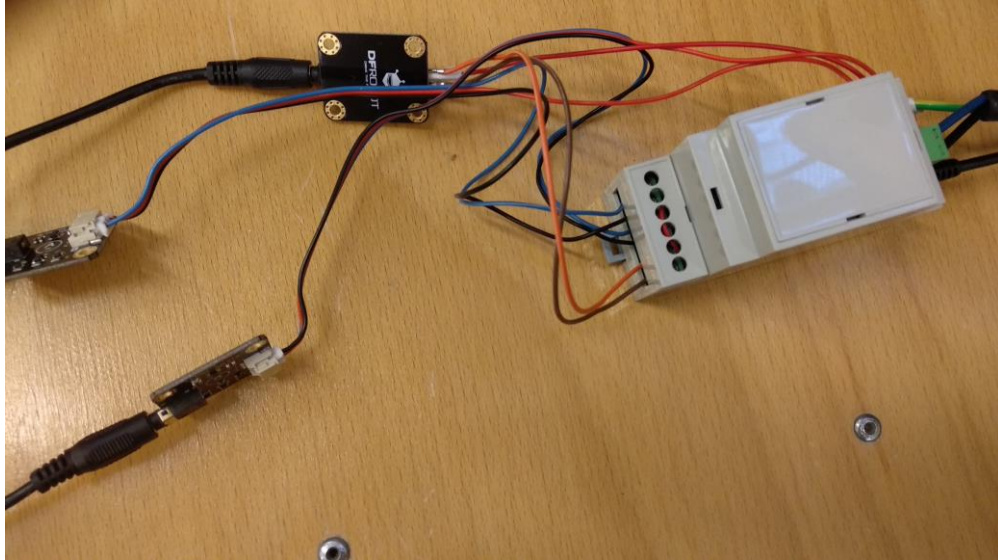


Figura 3.8 Dispositivo IoT basado en Arduino

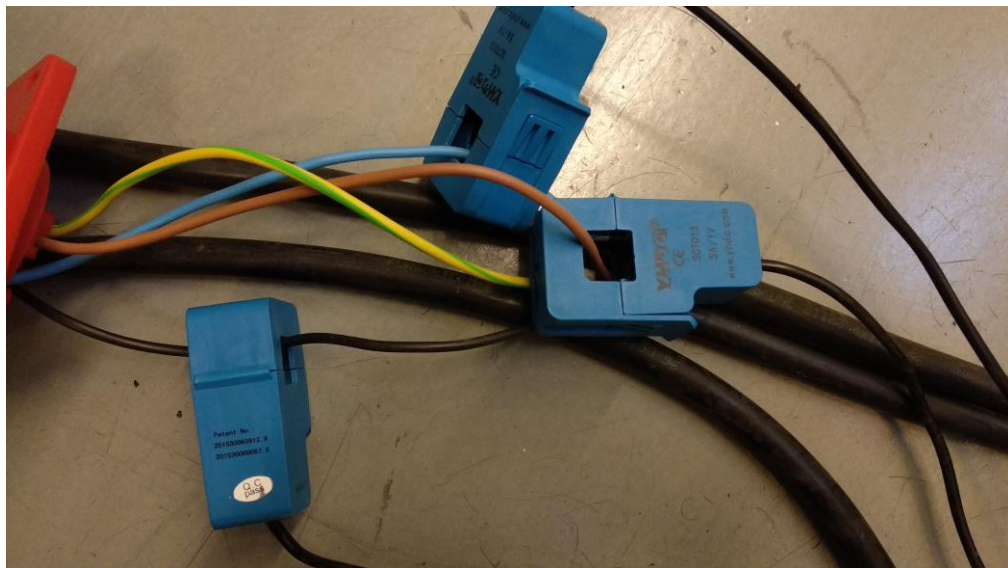


Figura 3.9 Pinzas amperimétricas del dispositivo IoT

La figura 3.9 muestra el montaje sobre el motor para llevar a cabo las mediciones

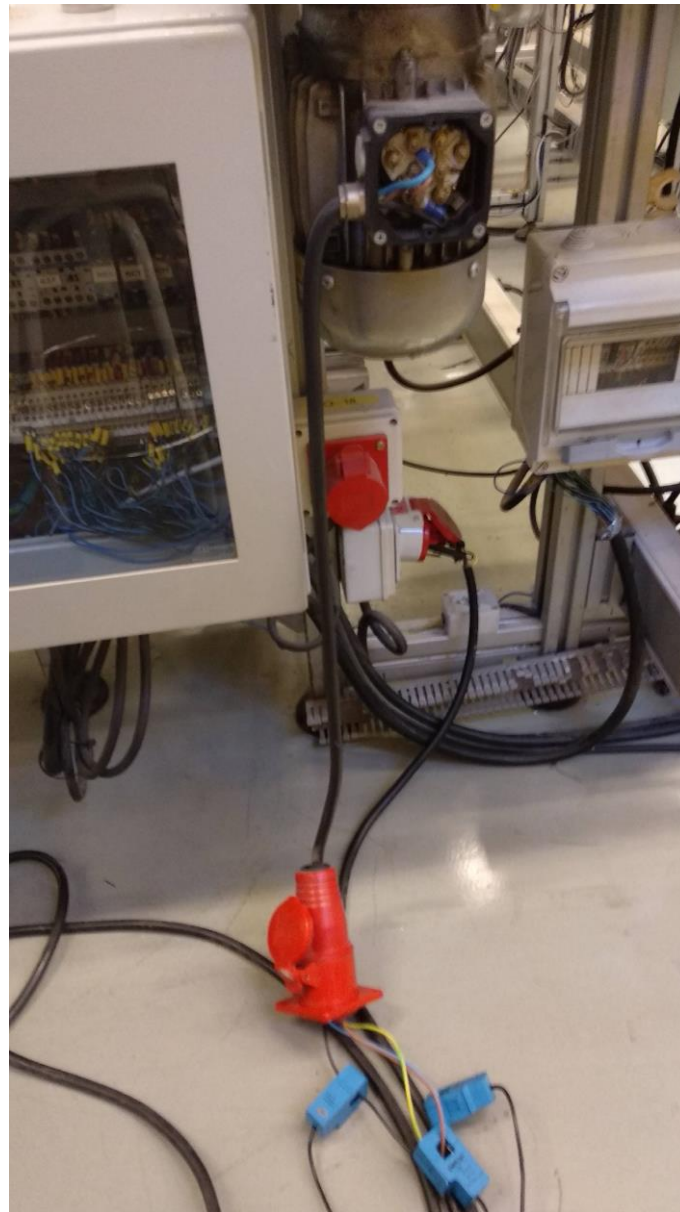


Figura 3.10 Montaje para medición del motor del conveyor

3.4 Acondicionamiento y procesamiento de datos

En este apartado, se presenta por separado el tratamiento de datos procedentes de los PLCs para el cálculo de tiempos, de los que almacena el dispositivo IoT para el modelo de novedad.

Para importar a Matlab los datos almacenados en InfluxDB, se hace uso de las librerías abierta, influx-Matlab⁹.

3.4.1 Cálculo de tiempos

Puesto que cargar al PLC con la tarea de medir tiempos para cada estado podría llegar a afectar a la capacidad del proceso, previamente se han registrado los estados de accionamiento digitales, midiendo las diferencias de tiempos entre los cambios de estados del proceso, podemos medir los tiempos de desplazamiento de las piezas y espera, así como los tiempos que una estación está libre.

El bucle siguiente, genera una timetable de calendarDuration con los tiempos que dura cada estado hasta su cambio guardado históricamente al principio de su activación.

```
p1=zeros(42,1); p2=zeros(42,1);
i=1;
for k=2:length(mytable{:, :})
    for j=1:42
        if mytable{k,j} ~= mytable{k-1,j} && mytable{k,j} == 1
            p1(j) = k;
        elseif mytable{k,j} ~= mytable{k-1,j} && mytable{k,j} == 0
            p2(j) = k;
        end
        if p1(j) > 0 && p2(j) > 0
            tiempos{i,j} = between(mytable.time(p1(j)),mytable.time(p2(j)));
            a(i)=mytable.time(p1(j));
            tiempos.Time = a;
            p1(j) = 0; p2(j) = 0;
            i=i+1;
        elseif k == length(mytable{:,j}) && (p1(j) == 0 || p2(j) == 0)
            p1(j) = 0; p2(j) = 0;
        end
    end
end

e=split(tiempos{:, :}, 't');
v=seconds(e);
```

⁹

Sala,E.

Influx-Matlab.

GitHub,2018,.

Obtenido de <https://github.com/EnricSala/influxdb-matlab>

En la timetable, mytable, se han escrito previamente datos leídos de InfluxDB con los estados del proceso. Cabe destacar que acabado el bucle, deben transformarse los calendarDuration en valores de tiempo, véase las funciones Split y seconds, para poder escribirlos posteriormente en una nueva timetable para escribir los tiempos calculados en la timetable.

3.4.2 Modelo de novedad

De cara a definir el modelo de novedad sobre el comportamiento del motor conductor de los conveyors, el registro de datos cuenta con datos de consumo diferenciados de un régimen de trabajo sano y régimen de trabajo no sano, es decir fuera de lo esperado

Para proceder a entrenar el modelo de novedad, se acondiciona la información sana, según los siguientes pasos:

- Se dividen las medidas en grupos con tantas medidas como dure un ciclo de proceso
- Para cada grupo de medidas, y para cada fase del motor, se definen los indicadores estadísticos: media, valor máximo, desviación estándar, varianza, factor de forma, factor de cresta, factor de impulso, asimetría estadística y curtosis
- Se normalizan todas las medidas
- Se genera una matriz de reducción de dimensiones a partir de estas medidas normalizadas para reducir de treinta a dos, el número de dimensiones
- Se usa la matriz con las medidas normalizadas sanas para trabajar con dos dimensiones
- Se divide el conjunto bidimensional de medidas sanas para tener medidas para entrenamiento y verificación a posteriori
- Se entrena el modelo, mediante la función de entrenamiento de modelo de máquina de vectores de soporte con las medidas para entrenamiento.

A continuación, se acondiciona la información no sana, para ello se siguen los pasos:

- Se dividen las medidas en grupos de duración equivalente a un ciclo de proceso
- Para cada grupo de medidas, y para cada fase del motor, se calculan los mismos indicadores estadísticos mencionados previamente
- Se normalizan todas las medidas respecto la normal y μ de los datos sanos
- Se usa la matriz de reducción de dimensiones definida previamente para reducir de treinta a dos, el número de dimensiones.

Una vez está toda la información acondicionada, se aplica la función predict en función los vectores de soporte definidos previamente para evaluar la información sana y no sana para verificación.

El resultado de la verificación se ilustra gráficamente en la figura 2.8.

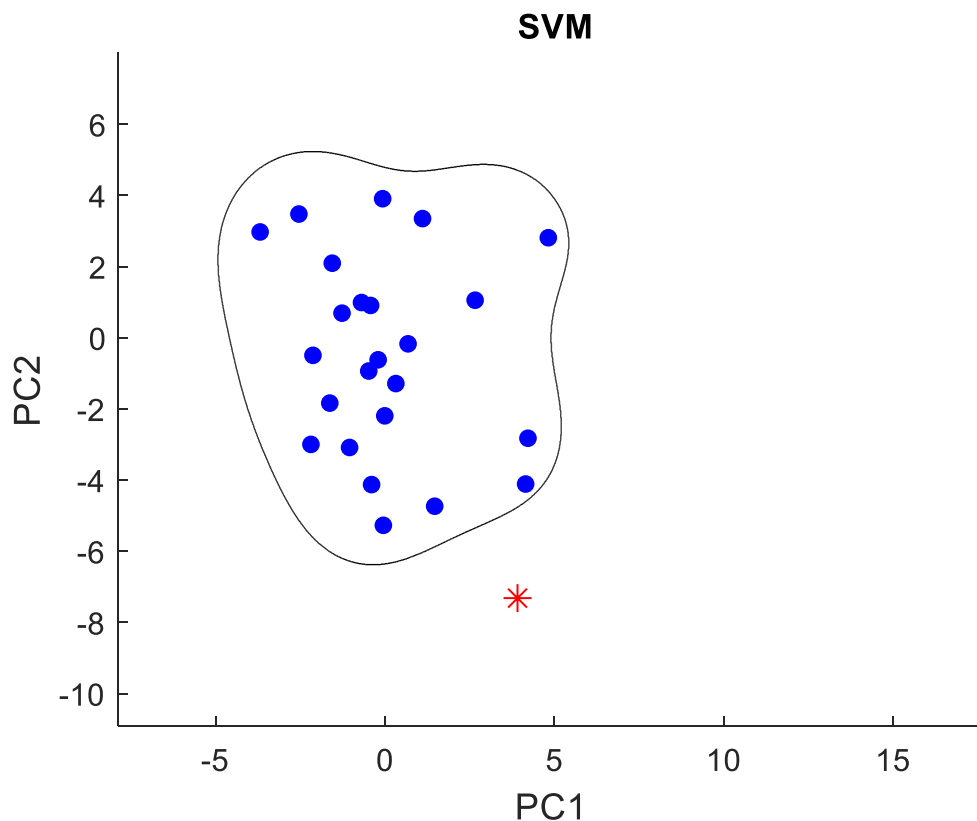


Figura 3.11 Modelo de novedad. Representación de la evaluación de medidas 1

En la figura 2.8, vemos delimitada el área que define comportamientos sanos, los puntos representan medidas acordes con un comportamiento sano para el modelo de novedad, la estrella roja se corresponde con un comportamiento no reconocido y por descarte no sano.

Sin embargo, de los 25 puntos representados, se conoce que 20 medidas se corresponden con comportamientos no correctos, ya que durante la toma de medidas se aumentó en gran medida la carga desplazada por los conveyors.

Si bien a priori se podría concluir que el modelo no funciona correctamente, si se repasan las medidas tomadas como comportamiento no sano, apenas existen centésimas de diferencia entre medidas sanas y no sanas.

Se atribuye tal comportamiento a la posibilidad de que el motor de estas líneas esté sobredimensionado, ya que las variaciones de carga sobre el conveyor no tuvieron ningún efecto sobre el motor.

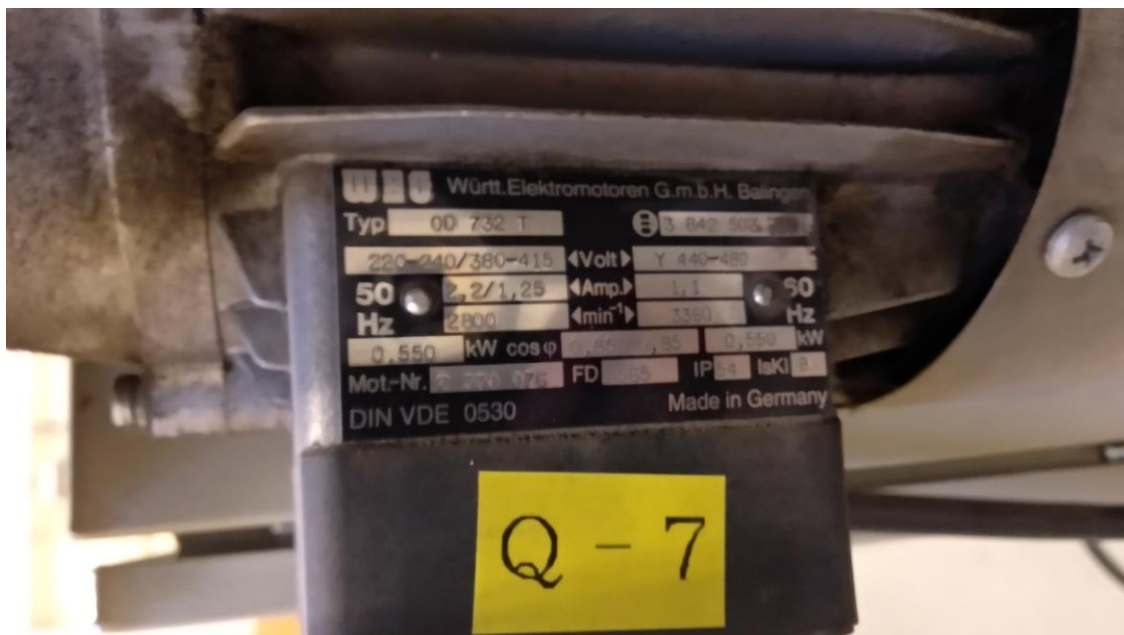


Figura 3.12 Placa de identificación del motor tractor del conveyor

Dicho esto, tratándose este, de un caso de estudio, se añade a todas las medidas de corriente no sanas 0,2A y se repite el acondicionamiento y evaluación de la información, el resultado gráfico se ilustra en la figura 2.9.

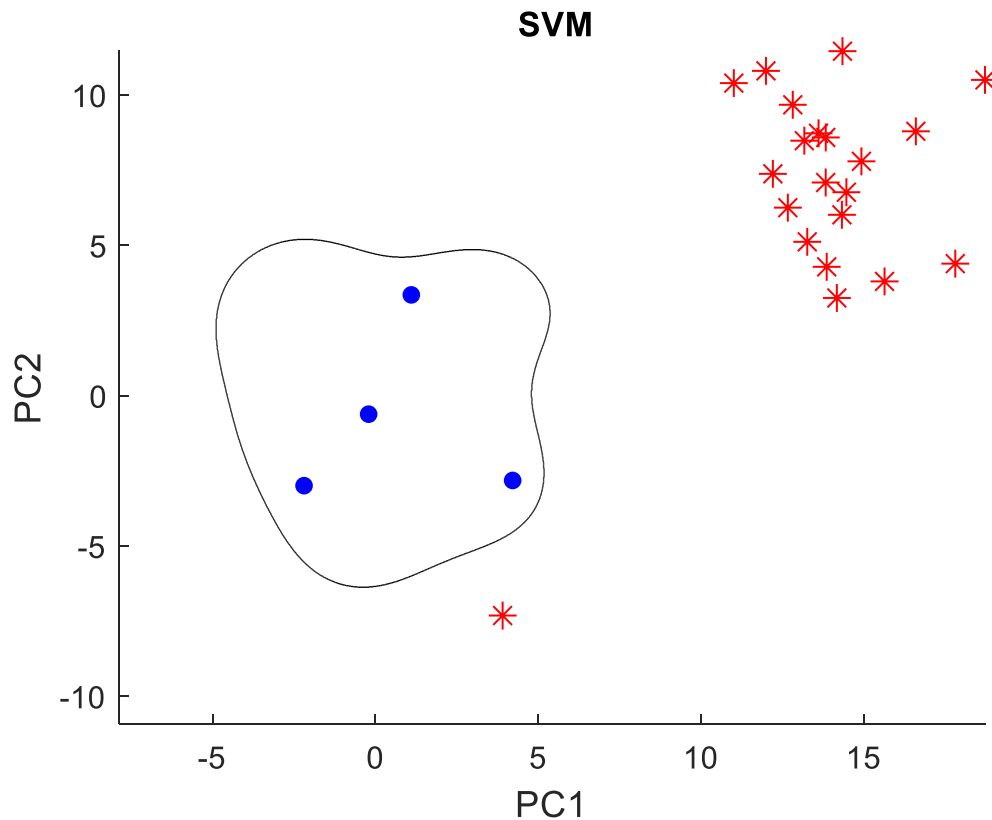


Figura 3.13 Modelo de novedad. Representación de la evaluación de medidas 2

Finalmente, la figura 2.9 ilustra que, 21 de las 25 medidas clasificadas como no sanas, dato acorde con lo esperado. Por lo tanto, solo un 4% de las verificaciones ha sido errónea.

Una vez el modelo es funcional y verificado, se almacena los resultados de clasificación del modelo en la base de datos.

3.5 Visualización

3.5.1 Preparación

Para preparar los dashboards, hace falta añadir a Grafana las bases de datos que almacenan la información, esto se hace desde la configuración del cliente.

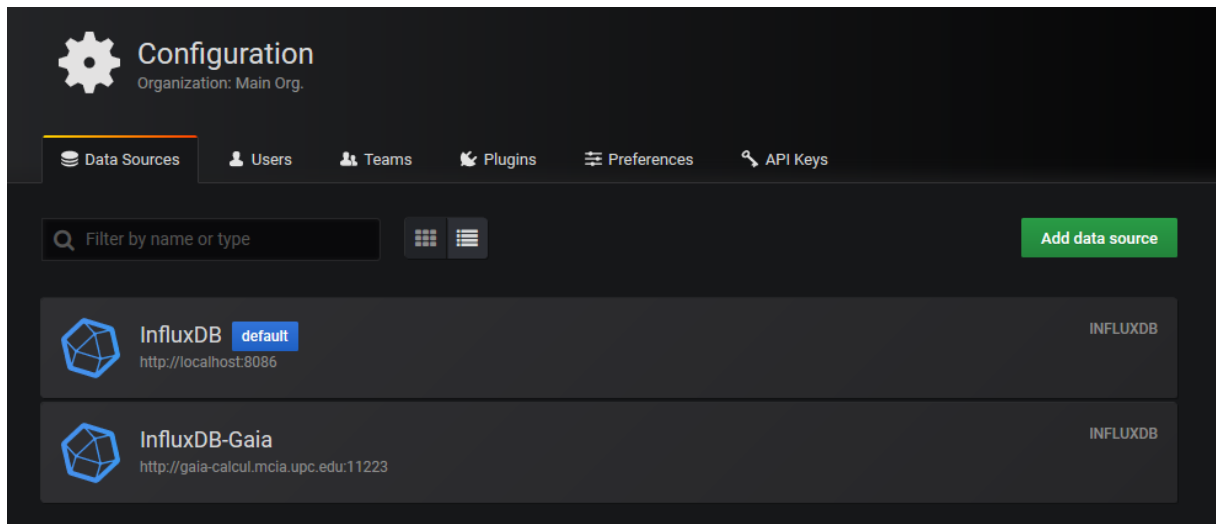


Figura 3.14 Definición de DB en Grafana

En la figura 3.10, puede ver como se integran tanto la base de datos local como la remota alojada en uno de los equipos del edificio Gaia.

3.5.2 Dashboards

Los dashboards finales obtenidos figuran a continuación, el dashboard de producción y mantenimiento, lo ilustra la figura 3.11.

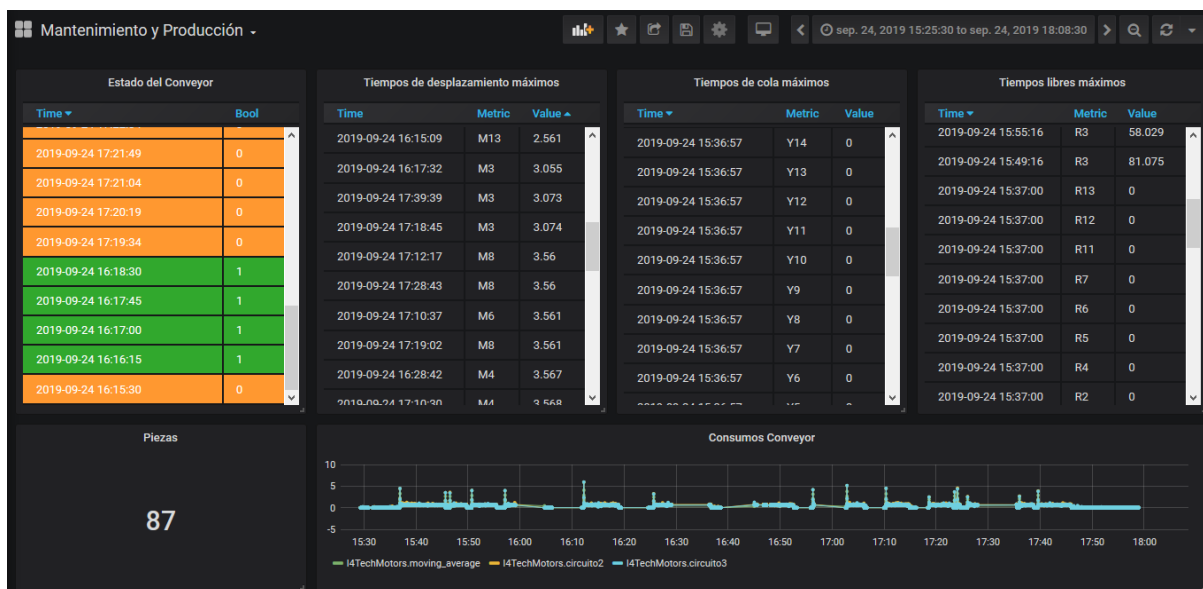


Figura 3.15 Dashboard de mantenimiento y producción

Este registro recopila el histórico de diagnósticos del conveyor, alertando mediante un código de colores sobre el correcto o incorrecto funcionamiento del mismo.

Time ▼	Bool
2019-09-24 17:22:34	0
2019-09-24 17:21:49	0
2019-09-24 17:21:04	0
2019-09-24 17:20:19	0
2019-09-24 17:19:34	0
2019-09-24 16:18:30	1
2019-09-24 16:17:45	1
2019-09-24 16:17:00	1
2019-09-24 16:16:15	1
2019-09-24 16:15:30	0

Figura 3.16 Registro de diagnósticos del modelo de novedad

Los registros de movimiento, cola y no ocupación, ofrecen información de los desempeños más críticos durante el día.

Tiempos de desplazamiento máximos			Tiempos de cola máximos			Tiempos libres máximos		
Time ▼	Metric	Value	Time ▼	Metric	Value	Time ▼	Metric	Value
2019-09-24 17:18:45	M3	3.074	2019-09-24 15:36:51	Y13	1	2019-09-24 17:38:20	R3	59.905
2019-09-24 17:14:28	M9	125.982	2019-09-24 15:36:51	Y12	1	2019-09-24 17:35:22	R14	58.709
2019-09-24 17:12:48	M2	225.934	2019-09-24 15:36:51	Y11	1	2019-09-24 17:34:33	R9	23.322
2019-09-24 17:12:39	M1	9.608	2019-09-24 15:36:51	Y10	1	2019-09-24 17:33:34	R8	52.832
2019-09-24 17:12:17	M8	3.56	2019-09-24 15:36:51	Y9	1	2019-09-24 17:32:11	R10	6.592
2019-09-24 17:11:50	M13	2.557	2019-09-24 15:36:51	Y8	1	2019-09-24 17:30:16	R9	253.63
2019-09-24 17:10:37	M6	3.561	2019-09-24 15:36:51	Y7	1	2019-09-24 17:23:10	R9	52.789
2019-09-24 17:10:30	M4	3.568	2019-09-24 15:36:51	Y6	1	2019-09-24 16:29:39	R8	2430.737
2019-09-24 16:28:42	M4	3.567	2019-09-24 15:36:51	Y5	1	2019-09-24 16:26:59	R8	69.492
2019-09-24 16:26:22	M1	10.70				2019-09-24 15:55:16	R3	58.029

Figura 3.17 Registros de tiempos del proceso

Un registro de piezas indica el número de piezas producido hasta el momento.

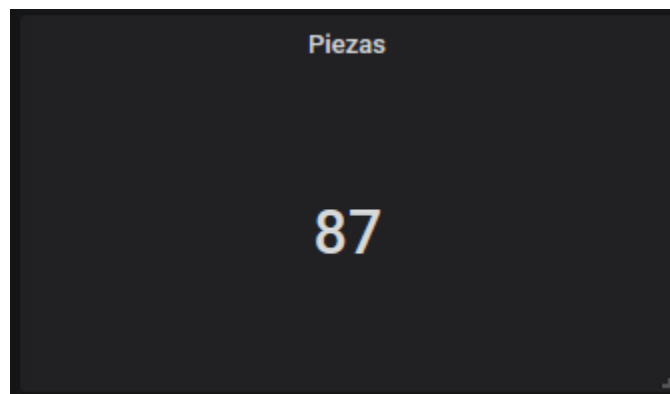
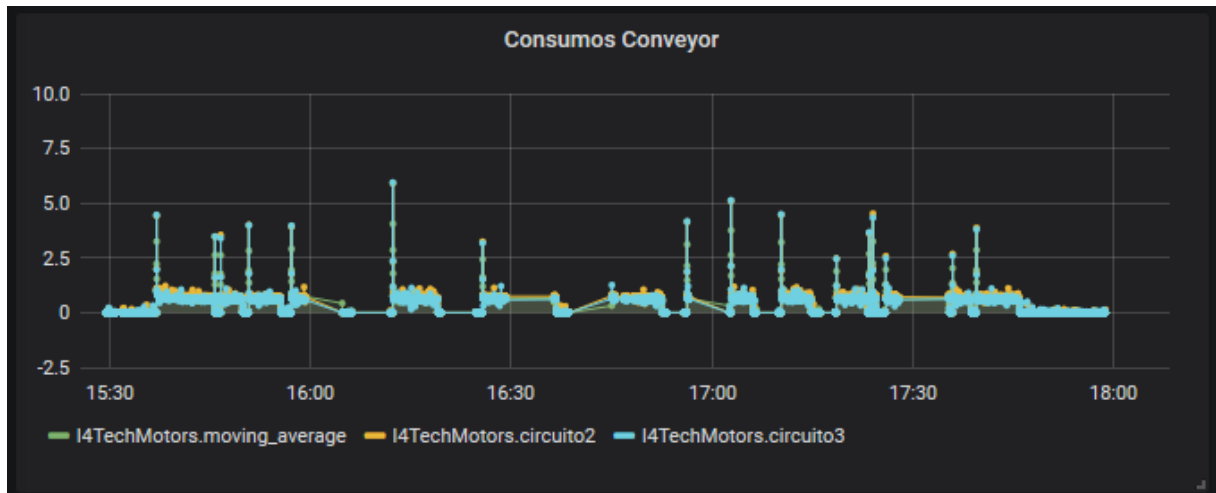


Figura 3.18 Contador de piezas



Puede consultarse los consumos del motor tractor del conveyor.

Figura 3.19 Histórico de consumos del motor tractor del conveyor

Adicionalmente, también se integran los dashboards desarrollados en Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de pieza ¹⁰



Figura 3.20 Dashboard de consumo

¹⁰ Pumares Cerezo, J. Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de pieza. Trabajo de fin de grado, UPC, 2019.

4 CONCLUSIONES

El principal objetivo de este estudio era implementar una solución de supervisión basada en inteligencia artificial orientada al mantenimiento predictivo, así como la integración vertical de los niveles de automatización necesarios para tal propósito.

El caso de estudio, desde el principio planteaba, en la medida de lo posible, ser lo más semejante posible a un caso práctico de implementación industrial. Estableciendo restricciones como la posibilidad de cambiar la máquina, programa y/o secuencia.

La mayor dificultad que ha planteado este trabajo, ha sido enlazar y aprender sobre los distintos campos tratados, así como asimilar e integrar trabajos realizados por alumnos previos.

El conocimiento adquirido para trabajar en los campos de la adquisición, almacenamiento y tratado de datos, así como el de la IA, no solo me ha permitido alcanzar las competencias para llevar este trabajo a su fin, si no que además me ha habilitado como interlocutor con perfiles orientados a la ingeniería de datos.

De cara a un trabajo futuro, existen distintas propuestas que podrían enriquecer un estudio como el llevado a término:

- Añadir dispositivos de medida adicionales al equipo, como pueden ser medidores de vibraciones, para detectar anomalías imperceptibles desde el punto de vista de consumos. Esta adición supondría una mejora directa, ya que, como se comentaba en el caso de las mediciones defectuosas, si bien aplicando carga adicional, la bancada de la línea se movía, debido al sobredimensionado del motor, tal anomalía no era perceptible vía consumos.
- Reconocer y clasificar tipos de fallo, en este trabajo, la implementación del modelo de IA se ha limitado a diferenciar entre estado sanos y no sanos del proceso. No obstante, puede ser interesante de cara al mantenimiento, poder clasificar causas de comportamiento no sano para ofrecer una mejor cobertura de diagnóstico, así como potencialmente reducir los tiempos de mantenimiento.
- Automatizar el procesado de datos, esta es una mejora sencilla, pero que aportaría un alto valor añadido, ya que los códigos de procesado de datos ejecutados en Matlab y debería reescribirse la lectura de medidas de la base de datos para nuevos casos.

- Implementar el modelo de IA en controladores dedicados a pie de máquina, de modo que el procesado no se ejecute de modo remoto, si no en la propia línea, dando pie a que la única información necesaria a adquirir o almacenar sean directamente los informes de diagnóstico.

Cierro el trabajo con unos resultados que valoro como satisfactorios, estos han superado mis expectativas y me dejan con ganas de conocer más retos en esta dirección, ya que, en un momento como el que nos encontramos, donde se están adoptando nuevas tecnologías y están reformulándose modelos de negocio, la implementación de una tecnología como es la IA, puede suponer para las empresas una oportunidad de progresar en el mercado o bien no quedarse atrás respecto a sus competidores.

5 BIBLIOGRAFÍA

Festo. Electric Automation, pág.3-4. Catálogo Promocional, 2019/06.
Obtenido de
https://www.festo.com/net/SupportPortal/Files/652002/ElectricAutomation_en.pdf

Industry 4.0. Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Industry_4.0

Influx. InfluxDB command line interface (CLI/shell). Obtenido de
<https://docs.influxdata.com/influxdb/v1.7/tools/shell/>

Grafana. Getting started. Obtenido de
https://grafana.com/docs/guides/getting_started/

Matlab. Machine Learning Challenges: Choosing the Best Model and Avoiding Overfitting. Whitepaper, 2016.

Node-Red.org. node-red-contrib-influxdb. 2019.
Obtenido de <https://flows.nodered.org/node/node-red-contrib-influxdb>

Node-Red.org. node-red-contrib-modbus tcp. 2018.
Obtenido de <https://flows.nodered.org/node/node-red-contrib-modbus tcp>

Node-Red.org. User Guide. Obtenido de <https://nodered.org/docs/user-guide/>

Pumares Cerezo, J. Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de pieza. Trabajo de fin de grado, UPC, 2019.

Ruiz Soto, L. Study of artificial intelligence tools applied to the monitoring of an electromechanical system. Trabajo de fin de grado, UPC, 2018.

Sala,E. Influx-Matlab. GitHub,2018. Obtenido de
<https://github.com/EnricSala/influxdb-matlab>

Vercruysse,J. Manufacturing Process automation by means of PLC's under the
Industry 4.0 framework. Trabajo de fin grado, UPC, 2019.